

Bidirectional Labeling and Registration Scheme for Grayscale Image Segmentation

Lei Ma, Xiao-Ping Zhang, *Senior Member, IEEE*, Jennie Si, and Glen P. Abousleman

Abstract—In this paper, we introduce a new image segmentation scheme that is based on bidirectional labeling and registration and prove that its segmentation performance is equivalent to that of the conventional watershed segmentation algorithm. The proposed bidirectional labeling and registration scheme, which we refer to as bidirectional labeling and registration scheme (BIDS), involves only linear scans of image pixels. It uses one-dimensional operations rather than the queues that are used in traditional segmentation algorithms, which are two-dimensional problems. BIDS also provides unique labels for individual homogeneous regions. In addition to achieving the same segmentation results, BIDS is four times less computationally complex than the conventional watershed by immersion technique.

Index Terms—Bidirectional labeling, catchment basin, gradient map, image segmentation, region merge, region registration, watershed.

I. INTRODUCTION

THE watershed is a concept in geography, referring to the natural pathways formed from water flow due to Earth's gravity. The potential of Earth's gravity drives the water downhill to form watersheds that segment the terrain into different regions. When the watershed technique is used to segment images, the potential of Earth's gravity is replaced by the gradients of image pixel intensity values, and the image (terrain) is segmented into regions according to homogeneity with respect to intensity values of pixels (water).

Watershed approaches [1], [2] have been developed and widely used for segmentation of digital images. Conventional watershed segmentation methods use local information to achieve global homogeneity, and they are more efficient as compared to methods such as region growing and tree search. However, the watershed is more complicated than some other segmentation methods such as thresholding, but with better segmentation results in terms of homogeneity. In order to provide random access to the two-dimensional (2-D) pixel array, a queue [1] is used in many of the conventional watershed methods. Since the image gradient varies from image to image,

the length of the queue cannot be determined *a priori*, which can be up to the size of the image in worst-case scenario. This generally introduces uncertainty into memory management and calculation overhead, and may lead to difficulties in hardware-specific implementations.

Recent research related to watershed segmentation has focused on segmentation performance and speed on a variety of images. Potentials other than gradients have been used to segment images with complex content. For example, Hill *et al.* [3] introduced a novel concept of "texture gradient" to segment texture-rich images with conventional watershed segmentation routines. For color images, Ji and Park [4] exploited luminance and chrominance difference components to verify regional coherency. Shafarenko *et al.* [5] used color gradient in CIELUV space (LUV gradient) to provide both a color similarity measurement and a basis for watershed transformation.

As one of the best available image segmentation methods, the conventional watershed is very efficient in terms of computational complexity and memory consumption as compared to methods such as partial differential equations (PDE) [6] and Markov random fields (MRFs) [7]. Even so, research has focused on reducing the computational complexity to make real-time segmentation feasible in low-end industrial devices. To this end, Pitas and Cotsaces [8] used skeletonization by influence zone and the watershed transform to increase the efficiency of memory access, thus, greatly increasing the ability to segment large images. Moga *et al.* [9] reformulated the watershed transform for parallel machines, and to reduce the communication overhead, which significantly increased processing speed. Heuristic methods were also used to reduce complexities involved with the queue operation, including gradient modification and static queues [10], [11].

These modifications have improved watershed segmentation performance in different types of images and in different applications, but most of these algorithms have retained pixel random access in the original image lattice. Thus, the segmentation remains a 2-D problem, which makes these methods computationally complex.

In this paper, we provide a systematic method of making the watershed operation a deterministic process by using a bidirectional labeling and registration scheme (BIDS). The bidirectional labeling process alleviates the problems caused by large flat areas in the gradient map of the image, and the registration operation eliminates the need for a queue. By changing the original image lattice to a raster-scan order, BIDS makes use of one-dimensional (1-D) data only. The proposed method generates a catchment basin and watersheds that are equivalent to those generated by the conventional watershed by immersion

Manuscript received November 20, 2002; revised October 30, 2004. This work was supported in part by General Dynamics C4 Systems, in part by the National Science Foundation under Grants ECS-9553202 and ECS0002098, in part by the Canadian NSERC under Grant RGPIN239031, and in part by MICRONET. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aly A. Farag.

L. Ma and J. Si are with the Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: leima@asu.edu; si@asu.edu).

X.-P. Zhang is with the Department of Electrical Engineering, Ryerson University, Toronto, ON M5B 2K3 Canada (e-mail: xzhang@ee.ryerson.ca).

G. P. Abousleman is with General Dynamics C4 Systems, Scottsdale, AZ 85257 USA (e-mail: glen.abousleman@gdds.com).

Digital Object Identifier 10.1109/TIP.2005.857277

technique. The computational complexity of BIDS is linear, and is four-times less than watershed by immersion segmentation.

The remainder of this paper is organized as follows. In Section II, BIDS is developed, and its equivalence to watershed segmentation is proved. In Section III, detailed implementation steps for BIDS are presented, and computational complexities for both BIDS and conventional watershed by immersion are analyzed. Section IV provides experimental results, and a conclusion is given in Section V.

II. BIDIRECTIONAL LABELING AND REGISTRATION SCHEME

BIDS consists of a gradient calculation, gradient sorting, and drowning steps that are the same or similar to watershed by immersion. In BIDS, bidirectional labeling and region index set registration replace initialization, skeletonization (e.g., skeletonization by influence zone), and finalization steps in watershed by immersion.

A. Notation, Terms, and Definitions

The notation, terms, and definitions used in this paper are defined as follows. An image to be segmented is denoted by \mathcal{I} . $I_{i,j}$ denotes a pixel or a pixel value in image \mathcal{I} at location (i,j) . The gradient map is denoted as \mathcal{G} , and its magnitude at (i,j) is denoted by $G_{i,j}$.

Definition 1: A pixel's 8-connected neighbors, $\mathcal{NB}(I_{i,j})$, are the eight adjacent pixels in the horizontal, vertical, and two diagonal directions. This is illustrated below, where $I_{i,j}$ is the current pixel

$$\mathcal{NB}(I_{i,j}) = \begin{bmatrix} I_{i-1,j-1} & I_{i-1,j} & I_{i-1,j+1} \\ I_{i,j-1} & I_{i,j} & I_{i,j+1} \\ I_{i+1,j-1} & I_{i+1,j} & I_{i+1,j+1} \end{bmatrix}.$$

In the following, 8-connected neighbors are also referred to as connected pixels.

Definition 2: The sortmap is the ordered gradient map by pixel gradient magnitude values. Should the situation occur where some pixels have the same gradient magnitudes, these pixels are sorted by a certain raster order (e.g., row by row).

Definition 3: A downstream path P is a sequence of connected pixels, with their gradient magnitudes ranked in a steepest descent, monotonic, and nonincreasing order.

Definition 4: A (regional) minimum M in a gradient map \mathcal{G} is a pixel from which no downstream path leads to any other pixel in \mathcal{G} with lower gradient values.

Definition 5: A minimum set \mathcal{M} is a connected set of minima in Definition 4 with gradient value g .

Definition 6: An influence zone associated with a minimum M is a set of pixels, including M , that has a downstream path to the minimum M . An influence zone is the union of the catchment basin and the watershed associated with M , which are defined next.

Definition 7: Similar to [1], a catchment basin $C(M)$ associated with a minimum M is a set of pixels p in the lattice of image \mathcal{I} , such that a water drop at p flows down along the relief, following a certain downstream path P to M . The lines that separate different catchment basins are called watersheds. Note that, in this paper, the catchment basin and the watershed associated with M make up the influence zone associated with M .

Definition 8: A critical area is a connected set of pixels that have the same gradient values, such as that in basin and saddle areas.

B. Conventional Watershed by Immersion Method

The conventional watershed method [1] is reviewed briefly in this section. Conventional watershed segmentation includes the following procedures: gradient generation, drowning, and sorting. For immersion, the generation of catchment basins and influence zones are based on the geodesic distance and skeleton concept. Let $T_h(I)$ denote the pixel locations with values less than h , then the subset of the catchment basin $C(M)$, at level h , $C_h(M)$, is defined as

$$C_h(M) = \{p \in C(M), I(p) \leq h\} = C(M) \cap T_h(I) \quad (1)$$

where p a pixel in the image. The immersion process starts from the lowest water level $T_{h_{\min}}(I)$ and works its way up in h value.

If Y is a connected component of water level, $T_{h+1}(I)$, the relationship between Y and the current catchment basin formed by the previous water level $T_h(I)$ is examined according to one of the following three cases to determine the new subset of catchment basin at water level $T_{h+1}(I)$. This process is usually repeated several times to ensure that the pixels go to the nearest minimum M in the geodesic distance sense.

- Case 1) If Y does not contain any part of $C_h(M)$, then Y itself and the pixels enclosed by it form a new watershed.
- Case 2) If Y contains or connects with one connected $C_h(M)$, then the new set of pixels in $Y \cap T_h(M)$ forms the updated subset of catchment $C_{h+1}(M)$ at level $T_{h+1}(I)$.
- Case 3) In the case where Y contains or connects with multiple connected subsets $C_h(M_i)$, the components of Y are placed in the closest subset $C_h(M_i)$ in a geodesic distance sense.

In the critical areas, first-in-first-out (FIFO) queue structures are used to search and determine new catchment basins. There are uncertainties in the calculation since the sizes and shapes of the critical areas are unknown. The queue length and the number of times to dequeue these pixels are, therefore, undetermined for dynamically allocated queues. For statically allocated queues, the size can be up to the size of the image itself in the worst case. This makes it relatively hard to evaluate the computational complexity and adds possible run-time overhead into the calculation.

C. Novel Bidirectional Labeling and Registration Operations

BIDS makes use of some of the conventional watershed segmentation procedures such as gradient generation, drowning, and sorting, as many variants of watershed methods do. The novelties of BIDS are bidirectional sorting, sortmap bidirectional labeling, and registration. These procedures make BIDS a more efficient algorithm as we will show later. First, we introduce the critical components in BIDS.

- **Sorting.** The sorting process makes two passes through the image, and sorts the pixels by their gradient values according to a preselected sort order. As the name implies, there are two scan orders involved to generate two

sortmaps. In the proposed algorithm, the preselected orders are chosen to be 1) row by row, from left to right, and 2) row by row, from right to left. Other sort orders are also feasible that are based upon the presented principle. This sort order also determines the labeling directions within the critical area. Throughout the remainder of this paper, they are used interchangeably.

- **Bidirectional labeling.** Bidirectional labeling is a linear labeling process that only involves the current pixel and its 8-connected neighbors according to the labeling rules presented below with reference to Fig. 1.

Case 1) All pixels in the set $\mathcal{NB}(I_{i,j})$ are not labeled. The pixel will be assigned to a new region index, advanced by 1 from the current highest region index in the map. The first pixel being labeled will be given the region index “1” by default. In this case, the pixel has the smallest gradient value in the neighborhood.

Case 2) If the set $\mathcal{NB}(I_{i,j})$ is partially labeled, and all the labeled neighbors have the same region index, then the current pixel will be labeled as the same region index as the others in the neighborhood. This pixel has the highest gradient value among the pixels that are assigned to the region index, and it is on the downstream path to the same minimum.

Case 3) If the set $\mathcal{NB}(I_{i,j})$ is labeled, but with different region indices, the pixel is labeled as a region boundary pixel, and it separates different catchment basins.

Repeat these operations for each pixel in the image following the sort order, then each pixel in the image will have a region index indicating to which catchment basin it belongs.

- **Registration.** After labeling, the recursive registration process proceeds as follows. There are two labeling processes, each following a different labeling direction. For the two sets of region indices, $1, 2, \dots, N$, for labeling direction α and $1, 2, \dots, M$, for labeling direction β , let $L^i(p)$, $i = \alpha, \beta$ denote the region index for pixel p in the two region index sets, then for any nonregion boundary pixel p and any pixel p' in the image

if $L^\alpha(p) = L^\beta(p)$, then the region index is correct;
if $L^\alpha(p) < L^\beta(p)$, then

$$\forall p', \text{ if } L^\beta(p') = L^\beta(p), \quad \text{then assign } L^\beta(p') \leftarrow L^\alpha(p) \quad (2)$$

if $L^\alpha(p) > L^\beta(p)$, then

$$\forall p', \text{ if } L^\alpha(p') = L^\alpha(p), \quad \text{then assign } L^\alpha(p') \leftarrow L^\beta(p). \quad (3)$$

Note that, in the registration process, we choose not to use nonregion boundary pixels because the directions of the two labeling processes are different, and the region indices on the region boundary pixels may be ambiguous. For this purpose, a boundary map is generated to indicate the pixels that are boundary pixels which should not be used for registration. Also, the actual watershed boundaries exist in both boundary maps, while the false boundaries are different in the two maps. The registration process is, thus, equivalent to finding the in-

tersection of the boundary maps generated by the two labeling processes.

To implement the registration process, a hash table is first built for each set of region indices. The new indices are initialized incrementally starting from 1, following one of the labeling orders, which is denoted *order A*. Each pixel in the image is then traversed following the *order A*. At each pixel location, the new indices in the two hash tables are updated to the smaller of the two. The resulting new indices in the hash table for the *order A* is the consolidated region index set.

The consolidated region index set generated by registration is then used to relabel the image. The pixels in the resulting image will each have a unique region index indicating the homogeneous region to which it belongs.

The registration algorithm is shown in Algorithm 1. From the pseudocode, “labelmap” is used to denote the consolidated final label map. The term “tableA” denotes the hash table for *order A*, and “tableB” denotes the hash table for *order B*:

Data: counter = 0

Data: tableA

Data: tableB

Result: labelmap

initialization;

for all pixel $I_{i,j}$ that is not boundary pixel **do**

if tableA($I_{i,j}$) = 0 **and** tableB($I_{i,j}$) = 0 **then**

 tableA($I_{i,j}$) \leftarrow counter;

 tableB($I_{i,j}$) \leftarrow counter;

 counter \leftarrow counter + 1;

else if tableA($I_{i,j}$) = 0 **and** tableB($I_{i,j}$) \neq 0 **then**

 tableA($I_{i,j}$) \leftarrow tableB($I_{i,j}$);

else if tableA($I_{i,j}$) \neq 0 **and** tableB($I_{i,j}$) = 0 **then**

 tableB($I_{i,j}$) \leftarrow tableA($I_{i,j}$);

else if tableA($I_{i,j}$) > tableB($I_{i,j}$) **then**

 tableA($I_{i,j}$) \leftarrow tableB($I_{i,j}$);

else if tableA($I_{i,j}$) \leq tableB($I_{i,j}$) **then**

 tableB($I_{i,j}$) \leftarrow tableA($I_{i,j}$);

end

for all pixel $I_{i,j}$ **do**

 labelmap($I_{i,j}$) \leftarrow tableA($I_{i,j}$);

end

D. Equivalence of Bids to Watershed by Immersion

To systematically prove the equivalence of BIDS to conventional watershed segmentation, we first consider the case where no critical area exists in the image. In this case, any order of labeling should generate real watershed boundary lines. Then, we will consider a more general case where critical areas do exist. False boundary lines usually emerge as a result of a specific labeling direction, and separate the critical areas into “directional catchment basins.” We will prove that these directional catchment basins can be merged into real catchment basin with the registration operation.

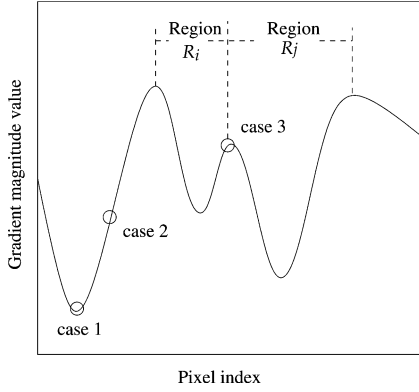


Fig. 1. Sortmap labeling rules.

Definition 9: A directional catchment basin is an area where a downstream path in the labeling direction exists for each pixel leading to its corresponding minimum M . This means that, for each downstream path in the directional catchment basin, namely the ascending pixel sequence $p_0, p_1, p_2, \dots, p_i, p_{i+1}, \dots, p_N$, the pixel p_i is always labeled before p_{i+1} .

Definition 10: A false boundary line is the line that exists in a real catchment basin, and separates it into directional catchment basins. This line appears because of the nature of the 1-D labeling direction, and the existence of critical area. False boundary lines in part define the boundary of a directional catchment basin.

1) *Equivalence Proof Under the Condition of Unique Gradient Magnitude Values:* We first prove that BIDS is equivalent to watershed by immersion when all of the gradient magnitude values in the image are different, and the critical areas do not exist.

Theorem 1: If the gradient map of an image satisfies the following conditions, then

- 1) each catchment basin has only one minimum;
- 2) for each point in a catchment basin, there exists a steepest downstream path that is *strictly* monotonically decreasing to the minimum, then we can conclude that any labeling direction generates segmentations in term of catchment basins satisfying Definition 7; in other words, the labeling rules of BIDS can generate segmentations of the image that are equivalent to conventional watershed segmentation [1], regardless of the labeling directions.

Proof: First, sort all pixels in a catchment basin by their gradient values $p_0, p_1, \dots, p_i, p_{i+1}, \dots, p_N$ in ascending order. Since p_0 has the lowest value, it will be labeled as the minimum for this catchment basin.

When pixel p_i is being labeled, the remainder of the pixels in its downstream path can be categorized into one of two sets: The labeled set P_i^l and the unlabeled set P_i^u . Due to the monotonically decreasing path, we have

$$\begin{aligned} p' < p_i, & \quad \forall p' \in P_i^l \\ p'' > p_i, & \quad \forall p'' \in P_i^u. \end{aligned} \quad (4)$$

Suppose there exists a strictly monotonic downstream path that is steepest from p_i to its minimum p_0 . Then, for pixel p_{i+1} , which is in the neighborhood of p_i , since

$$p_{i+1} \in P_i^u \quad (5)$$

therefore

$$p_{i+1} > p_i, \quad (6)$$

and

$$p_{i+1} \in \mathcal{NB}(p_i) \quad (7)$$

then, p_{i+1} will be assigned to the same label as p_i according to the labeling process presented in case 2 of the bidirectional labeling process. In other words, p_{i+1} is in the same catchment basin as p_i if there is no other pixel p_k where $k < i$ in the neighborhood of p_{i+1} .

If two pixels p_i and p_k in the neighborhood of p_{i+1} have downstream paths to different minima, and lower gradient values than p_{i+1} , then p_{i+1} will be labeled as a watershed. Therefore, between the two lower-valued pixels, there is no downstream path. ■

Corollary 1: If all gradient values in an image are unique, then the BIDS bidirectional labeling results are equivalent to those of watershed by immersion and do not depend on the labeling directions.

Proof: Since no pixels in the image have the same gradient value, it is obvious that each catchment basin has only one minimum, and the downstream path will be strictly monotonically decreasing. As Theorem 1 states, the results are equivalent.

2) *Equivalence Proof at the Presence of Critical Areas and False Boundary Lines:* In real-world images, critical areas do exist, especially when drowning is used to suppress noise in the image. Not only is the unique minimum of a catchment basin no longer guaranteed, but the downstream path may no longer be strictly monotonic. Within these critical areas, the bidirectional labeling direction will not follow the geodesic distance order. Rather, they follow one of the labeling directions. False boundary lines may appear depending upon the shapes of the critical areas.

If the critical areas in the bottom of the catchment basin are convex, then for any pixels in the area, there exists a line along the labeling direction such that all pixels on this line are inside the critical area. Thus, the pixels on the line are always labeled in a monotonic order along the labeling direction. When a pixel is to be assigned to a region index, the pixel before it on the same line has already been labeled. These two pixels will then have the same region index. Therefore, false boundary lines are not of concern in this case.

For critical areas that are concave, directional catchment basins are usually generated, as shown in Fig. 2. They are created by certain combinations of the labeling direction and the direction of the concaveness, which will be explored in more detail later in this section. Referring to Fig. 2, suppose areas B and C have the same gradient magnitude level, which is lower than that of A . Following labeling direction 1, when the pixel in the top left corner of area C is being labeled, there are no labeled pixels that are connecting it to the already partially labeled area B . Thus, according to the labeling rules in BIDS,

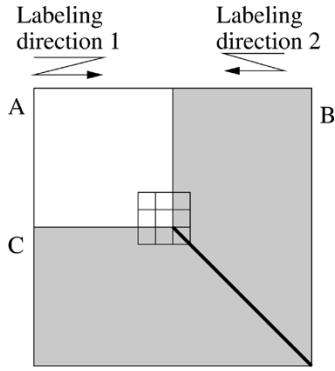


Fig. 2. For a given labeling direction (e.g., row by row, from left to right) and the direction in concaveness as shown, false boundary lines are generated within the same critical area since the top left pixel in C and its subsequent pixels in C are given to different labels than B , according to the labeling rules.

the pixels in area C will be given a new region index. When the two labeled areas meet, the bold line is created and is called the false boundary line, which separates the two directional catchment basins that are supposed to have the same label.

Note, however, that if the labeling direction is changed to direction 2, as shown in Fig. 2, the illustrated false boundary line under this condition will not appear. In this paper, we consider four types of possible concave orientations, which may result in false boundary lines if they resonate with some labeling directions. The false boundary lines can break catchment basins into directional catchment basins according to Definition 9.

Property 1: The directional catchment basins have the following properties.

- 1) Directional catchment basins are subsets of the real catchment basin that contains the critical area to which they belong.
- 2) Directional catchment basins that belong to the same catchment basin are nonoverlapping.

Proof: Since directional catchment basins are generated within a critical area separated by false boundary lines, and the critical area itself belongs to a catchment basin according to Definition 7, then the directional catchment basin is a subset of the real catchment basin. Each pixel in the image is labeled with one and only one region index in each labeling process; thus, the directional catchment basins cannot overlap.

Proposition 1: For a concave critical area located at the bottom of a catchment basin with one of the four directions shown in Fig. 3, there exist two labeling directions such that when used in labeling, these two directions generate different sets of directional catchment basins, or at least one of them does not generate directional catchment basins.

Proof: This can be proved by examining all of the possible combinations of labeling directions and orientations of the concave critical areas in the bottom of a catchment basin. Fig. 3 is an illustration of all possible cases that can indeed resonate and create false boundary lines. Since these critical areas are located at the bottom of a catchment basin, none of the pixels in the neighborhood of such a critical area are labeled before it is labeled. The directional catchment basins only depend on the

concaveness of the critical area with respect to the labeling directions. The generation of these false boundary lines are due to the fact that raster scan order is used rather than 2-D geometric distance.

To ensure that two labeling processes do not generate overlapping false boundary lines, the labeling directions are chosen according to Fig. 3 and correspond to different directional catchment basins, i.e., top down from left to right, and bottom up from right to left.

Proposition 2: For a critical area on the saddle position, there exist two labeling directions such that when used in labeling, these two directions generate different sets of directional catchment basins in this critical area.

Proof: For the critical area on the saddle position, i.e., the critical area next to another area that has a lower gradient value and is already labeled, false boundary lines will appear at the border of the critical regions and the already labeled area. In Fig. 4, all possible saddle critical area positions are illustrated. In this case, the generation of false boundary lines do not depend on the concaveness of the area shape. Rather, they are generated because there exists an adjacent area that is already labeled. This only happens when the labeling direction is going from the unlabeled critical area into the already labeled area. From the combinations in Fig. 4, there exist a set of labeling directions that can generate different directional catchment basins, i.e., top down from left to right, and bottom up from right to left. ■

Proposition 3: For a general critical area, there exist two labeling directions such that when used in labeling, these two directions generate different sets of directional catchment basins in this critical area.

Proof: The Proof of Proposition 3 follows from Proposition 1 and Proposition 2. Starting from the bottom critical area with lowest gradient value using Proposition 1, then considering the next lowest gradient valued critical area using both Proposition 1 and Proposition 2, we can iteratively label the entire critical area without overlapping false boundary lines. By carefully exploiting the combinations in Figs. 3 and 4, we conclude that one of the recommended set of labeling directions that would generate different sets of directional catchment basins is top-down from left to right, and bottom-up from right to left. ■

Using Algorithm 1, these different sets of directional catchment basins can be merged into a real catchment basin. The proof is given in Theorem 2.

Theorem 2: The directional catchment basins resulting from different labeling directions can be merged into a real catchment basin through recursive registration.

Proof: Let any region R_k with real region index $k = \{1, 2, \dots, K\}$ be assigned into two directional region index sets, $1, 2, \dots, N$, and $1, 2, \dots, M$, by the two labeling procedures, respectively. Let $L^\alpha(p)$ and $L^\beta(p)$ be the directional region index of pixel p by labeling direction α and β , respectively, R_n^α denote a set of connected pixels with the same region index n under labeling direction α and R_m^β denote a set of connected pixels with the same region index m under labeling direction β . Then, for a pixel $p \in R_k$, its region indices from the two labeling directions are $L^\alpha(p)$ and $L^\beta(p)$, and

$$\exists p', \text{ s.t. } L^\alpha(p) = L^\alpha(p') \quad \text{and} \quad L^\beta(p) \neq L^\beta(p'). \quad (8)$$

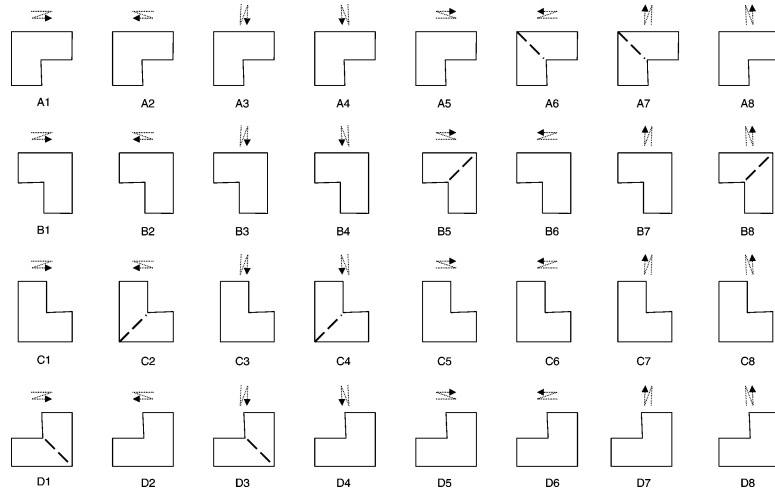


Fig. 3. Illustration of all possible combinations of labeling directions and the orientation of the concaveness of the critical areas. Eight cases are identified as resonant cases between the labeling direction and the orientation of concaveness, which can result in false boundary lines and consequently directional catchment basins. The arrows depict the labeling directions, and the bold dashed lines depict the false boundary lines.

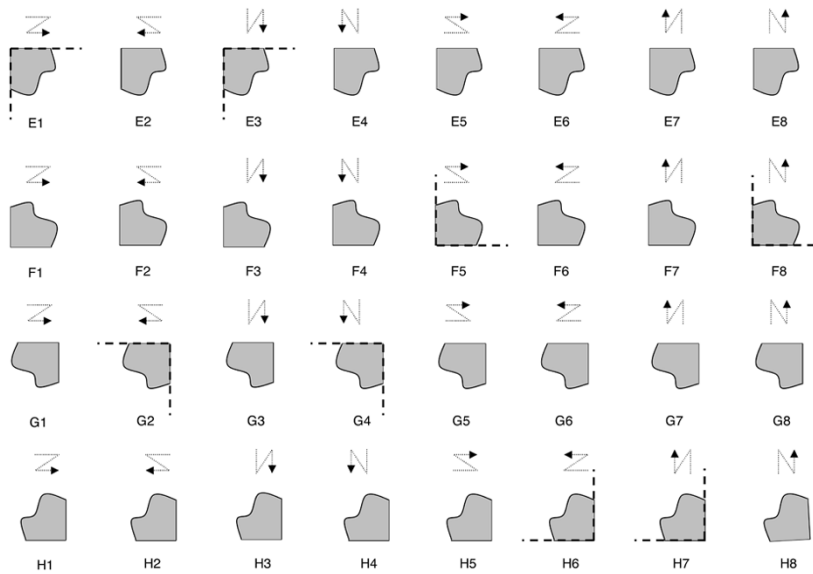


Fig. 4. Illustration of all possible combinations of saddle critical area positions and the labeling directions. The gray region denotes the position of already labeled area with lower gradient value. The white region denotes the saddle critical area position. The annotation at the top denotes the labeling directions, and the dashed lines denote false boundary lines.

Let

$$k = \min\{L^\alpha(p), L^\beta(p)\}, \quad \forall p \in R_k. \quad (9)$$

Without loss of generality, if

$$L^\alpha(p) = k, L^\beta(p) \neq k \quad (10)$$

then, by applying recursive registration in (2) and (3), p will be assigned to

$$L^\beta(p) = k. \quad (11)$$

Applying this to all p in the critical area, the entire critical area will be labeled by region index $L(p) = k$ or equivalently

$$\{p \mid L^\alpha(p) = i\} = \{p' \mid L^\beta(p') = j\} = \{p'' \mid L(p'') = k\}. \quad (12)$$

Thus, using region notation R , we have

$$R_i^\alpha = R_j^\beta = R_k. \quad (13)$$

The resulting region index set is the real region index set. The two directional catchment basins are merged into their union, R_k . This can be applied to all directional catchment basins in

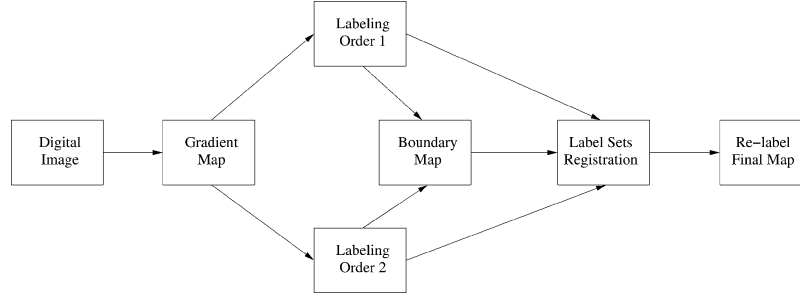


Fig. 5. Block diagram of proposed BIDS algorithm.

 TABLE I
 COMPARISON OF COMPUTATIONAL COMPLEXITY BETWEEN BIDS AND WATERSHED BY IMMERSION;
 BIDS SHOWS FOUR TIMES LESS COMPLEX THAN WATERSHED BY IMMERSION

Processing Methods	Bi-directional Labeling and Registration	Watershed by Immersion
Image input/output	I/O	I/O
Gradient calculation	$18N(A)$	$18N(A)$
Sorting	$N + D - 1(A), 2N(=)$	$N + D - 1(A), 2N(=)$
Initial labeling	-	$DN + 16N(><), 3N(=)$
SKIZ	-	$9DQ(A), 34DQ(=), 83DQ(><)$
Finalization	-	$N(A), 3N + 17DQ(=), DN + 8DQ + N(><)$
Bi-directional labeling	$6N(A), 37N(=), 17N(><)$	-
Edge mark	$34N(A), 5N(=), 13N(><)$	-
Registration	$N(A), 5N(=), 11N(><)$	-
Total	$I/O; 60N + K - 1(A); 49(=); 58N(><)$.	$I/O; 9DQ + 20N + k - 1(A); 7N + 34DQ(=); 2DN + 91DQ + 17N(><)$.
Approximation	$I/O + 167N + 255$ operations	$I/O + 706N + 255$ operations

the real catchment basin, which is the enclosure of the union of the constituting directional catchment basins. ■

As shown in the preceding discussion, BIDS can indeed generate segmentations according to the conventional watershed definition [1]. In the next section, we show that BIDS is much more efficient computationally than the conventional watershed by immersion algorithm.

III. IMPLEMENTATION OF BIDS AND COMPUTATIONAL COMPLEXITY ANALYSIS

The implementation of the BIDS algorithm starts with the gradient value calculation, followed by flooding. Bidirectional sorting, labeling, and registration procedures are carried out subsequently, as shown in Fig. 5.

A. Gradient Calculation

The gradient magnitude value of a pixel is calculated as [12]

$$G_{i,j} = \sqrt{G_{i,j}^x{}^2 + G_{i,j}^y{}^2}$$

where $G_{i,j}^x$ and $G_{i,j}^y$ are the gradient values along the two perpendicular axes defined and calculated by their corresponding gradient operators as

$$G_{i,j}^x = \langle \mathcal{I}, H^x \rangle_{i,j}, \quad G_{i,j}^y = \langle \mathcal{I}, H^y \rangle_{i,j}$$

where H^x and H^y are the gradient operators, and $\langle \cdot, \cdot \rangle$ denotes the convolution operation.

The gradient magnitude can also be replaced by other dissimilarity measurements for segmentations based on other homogeneous features such as texture or color.

B. Flooding and Sorting

The flooding operation is defined as

$$\bar{G}_{i,j} = \begin{cases} G_{i,j}, & \text{if } G_{i,j} \geq T_d \\ T_d, & \text{otherwise.} \end{cases} \quad (14)$$

where T_d is the flooding parameter. The flooding operation aids in the removal of noise and weak edges. When tested on the IEEE test image set, T_d has a fairly robust equilibrium at 39 for natural grayscale images and 24 for MRI images.

Details pertaining to sorting, linear labeling, and registration are discussed in Section II-C.

C. Computational Complexity Analysis

BIDS is a linear procedure. To calculate the complete region index map, the image is scanned through eight times. In each scan, calculation involves only the pixel itself and its 8-connected neighbors. Here, we denote the number of pixels in the image by N , the number of gradient magnitude levels by D , and the average queue length in the conventional watershed algorithm by Q . Since different computational operations take different clock times, we use $(=)$ to denote assignment operations, (A) to denote arithmetic operations, and $(><)$ to denote comparisons. (I/O) is used to denote an image Input/Output process.

The computational complexity analysis is summarized in Table I.

From the table, we see that BIDS requires approximately four-times fewer operations than watershed by immersion. In actual implementations, the arithmetic operations usually require more CPU clock cycles than comparisons and assignments. Without implementation optimization, BIDS is at least

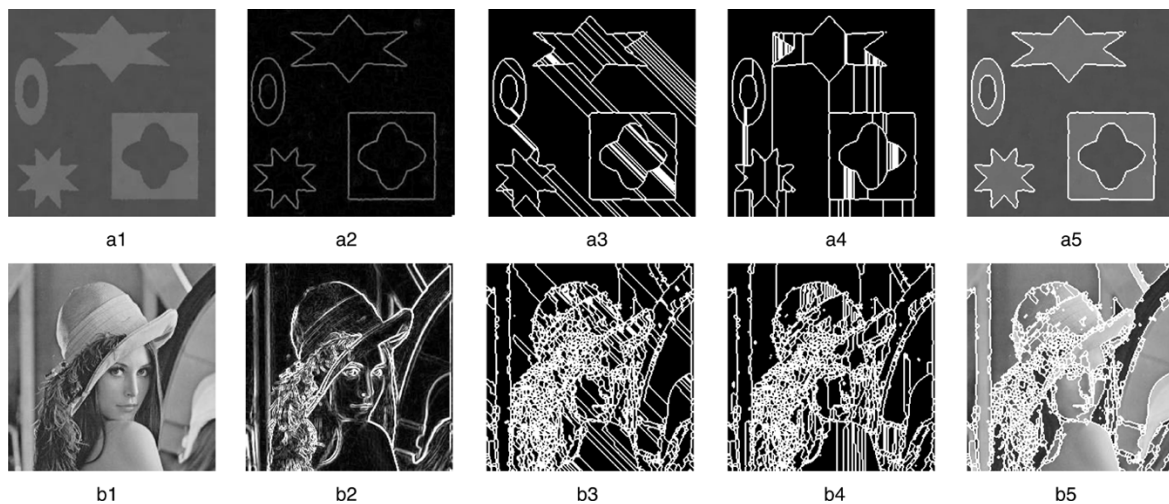


Fig. 6. BIDS applied to testing images with drowning threshold $T_d = 39$. (a1), (b1) Original image. (a2), (b2) Gradient map. (a3), (b3) Labeling row by row downward, from left to right. (a4), (b4) Labeling row by row upward, from right to left. (a5), (b5) Final segmentation by registration.

twice as fast, on average, than the optimal implementation of the conventional watershed that is presented in [10]. Note that heuristic preprocessing is used in [10] to accelerate the speed of the algorithm, which yields a slight deviation from the conventional watershed algorithm. This happens when drowning normalizes the gradient values, and some of the gradient levels are overlooked, which results in the absence of certain region boundary pixels. A comparison of BIDS without drowning to the watershed algorithm in [1] shows BIDS to be approximately four-times faster. Moreover, the conventional watershed method still has an uncertainty factor Q , which increases its complexity even further. Also, BIDS produces region index sets that can benefit from subsequent procedures that utilize the results from the segmentation.

IV. SIMULATION RESULTS

Experiments were carried out on several images to illustrate the performance of the BIDS algorithm. The algorithm was implemented in C++, and compiled with the GNU g++ compiler version 2.95.3–5, on a 1-GHz Pentium III machine without implementation optimization.

Fig. 6 shows the BIDS processing steps on two testing image. The image on the top row has large uniform intensity regions (critical areas), which are a major cause of false boundary lines. As shown in Fig. 6(a3), this particular labeling direction generates many false boundary lines in the critical area, while the other labeling direction [Fig. 6(a4)] does not. The drowning threshold used here was $T_d = 39$.

The set of images on the bottom row shows the BIDS processing steps for the Lena image. With a drowning threshold of $T_d = 39$, BIDS yields 807 segments for this image.

Without implicit implementation optimization, BIDS segments imagery of size 256×256 in an average of 0.27 s, which includes the gradient map (activity image) calculation. With 512×512 imagery, the run time averages approximately 1.10 s, which demonstrates the linearity of BIDS.

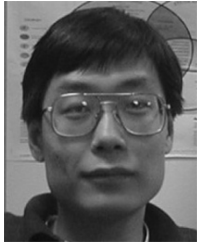
V. CONCLUSION

Watershed segmentation is a well established image segmentation method that has generated great interest to improve its performance further and to widen its application domain. In this paper, we have presented a novel segmentation method called the BIDS, which was compared to watershed by immersion segmentation. BIDS was shown to be more efficient and systematic than conventional watershed methods, while being easily implementable. We have shown in theoretical terms that BIDS generates uniquely labeled regions satisfying the definition of the catchment basin and watershed line in the original watershed by immersion technique without the use of queues. Computational complexity analysis shows that BIDS is a linear algorithm and requires four-times fewer operations than the conventional watershed algorithm. Simulation results also show that BIDS provides outstanding segmentation performance.

REFERENCES

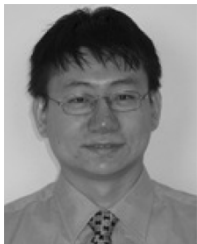
- [1] L. Vincent and P. Soille, "Watersheds in digital spaces—An efficient algorithm based on immersion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 583–598, Jun. 1991.
- [2] F. Meyer and S. Beucher, "Morphological segmentation," *Int. J. Vis. Commun. Image Rep.*, vol. 1, pp. 21–46, 1990.
- [3] P. R. Hill, C. N. Canagarajah, and D. R. Bull, "Texture gradient based watershed segmentation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 4, Orlando, FL, May 2002, pp. 3381–3384.
- [4] S. Ji and H. W. Park, "Image segmentation of color image based on region coherency," in *Proc. Int. Conf. Image Processing*, vol. 1, Chicago, IL, Oct. 1998, pp. 80–83.
- [5] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE Trans. Image Processing*, vol. 6, no. 11, pp. 1530–1544, Nov. 1997.
- [6] N. Paragios and R. Deriche, "Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision," *Int. J. Vis. Commun. Image Rep.*, vol. 13, no. 1/2, pp. 249–268, 2002.
- [7] R. Morris, X. Descombes, and J. Zerubia, "Fully bayesian image segmentation—An engineering perspective," in *Proc. 1997 Int. Conf. Image Processing*, vol. 3, Washington, DC, Oct. 1997, pp. 54–58.
- [8] I. Pitas and P. I. Cotsaces, "Memory efficient propagation-based watershed and influence zone algorithms for large images," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1185–1199, Jul. 2000.

- [9] A. Moga, B. Cramariuc, and M. Gabbouj, "An efficient watershed segmentation algorithm suitable for parallel implementation," in *Proc. Int. Conf. Image Processing*, vol. 2, Washington, DC, Oct. 1995, pp. 101–104.
- [10] P. D. Smet and R. Pires, "Implementation and analysis of an optimized rainfalling watershed algorithm," in *Proc. 12th Annu. Symp. Electronic Imaging: Science and Technology Conf.: Image and Video Communications and Processing*, San Jose, CA, Jan. 2000.
- [11] L. Ma, J. Si, and G. P. Abousleman, "Integrated and hierarchical sortmap-relabel image segmentation methods," *Opt. Eng.*, vol. 41, no. 11, pp. 2856–2865, Nov. 2002.
- [12] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.



Lei Ma received the B.S. degree in computer science from Truman State University, Kirksville, MO, in 1999. He is currently pursuing the M.S. degree at Arizona State University, Tempe.

His current research area is automatic target recognition and tracking in still image and videos.



Xiao-Ping Zhang (M'97–SM'02) received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1992 and 1996, respectively, all in electronic engineering.

From 1996 to 1998, he was a Postdoctoral Fellow at the University of Texas, San Antonio, and then at the Beckman Institute, University of Illinois, Urbana-Champaign. He held research and teaching positions at the Communication Research Laboratory, McMaster University, Hamilton, ON, Canada, in 1999. From 1999 to 2000, he was a Senior DSP

Engineer at SAM Technology, San Francisco, CA, and a Consultant at the San Francisco Brain Research Institute. Since Fall 2000, he has been on the faculty at Ryerson University, Toronto, ON, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. His current research interests include signal, image, and multimedia processing, as well as their applications in communications, biomedicine, and bioinformatics.

Dr. Zhang received the Science and Technology Progress Award from the State Education Commission of China for his significant contribution in a National High-Tech. Project in 1994. He is a Registered Professional Engineer in Ontario, Canada.



Jennie Si received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 1985 and 1988, respectively, and the Ph.D. degree from the University of Notre Dame, Notre Dame, IN.

She has been on faculty at Arizona State University, Tempe, since 1991, upon the completion of her Ph.D. degree, is now a Professor with the Department of Electrical Engineering. Her current research interests include theory and application of artificial neural learning systems, learning algorithms for statistical signal modeling, and data analysis. Her application

areas of interest include semiconductor process optimization and yield management; spatial-temporal biological signal modeling in motor cortical and auditory systems, 2-D visual information processing through region of interest and robust feature selection; and mining and interpretation of large data sets.

Dr. Si is a recipient of the 1995 NSF/White House Presidential Faculty Fellow Award and a recipient of the Motorola Excellence Award in 1995. She was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL in 1998 and 1999, and has been an Associate Editor of the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING since 1998 and the IEEE TRANSACTIONS ON NEURAL NETWORKS since 2001.



Glen P. Abousleman received the B.S. and M.S. degrees from The University of New Mexico, Albuquerque, in 1988 and 1990, respectively, and the Ph.D. degree from The University of Arizona, Tucson, in 1994, all in electrical engineering.

He was a Software Engineer and an Analog Design Engineer for Honeywell, Defense Avionics Systems Division, Albuquerque, from 1987 to 1988. In 1994, he joined the Satellite Communications Group at Hughes Space and Communications, El Segundo, CA, where he worked on satellite-based mobile communication systems.

In 1995, he joined the Speech and Signal Processing Laboratory, Motorola Integrated Information Systems Group, Scottsdale, AZ, where he worked on vocoder and image coder research and development. Currently, he is the Director of the Compression, Communications, and Intelligence Laboratory at General Dynamics C4 Systems, Scottsdale. He is also an Adjunct Professor with the Electrical Engineering Department, Arizona State University, Tempe. He has published over 75 technical articles in the areas of image processing/compression and wireless communications, and has filed 14 patents, of which seven have been issued. His research interests include multidimensional image compression and processing, wireless communication systems, intelligent systems, automatic target recognition, and neural networks.

Dr. Abousleman is a member of Eta Kappa Nu.